

AXN-SYMBOLON-SPEC v0.2 — The Symbolon as Split-Identity Verification Primitive: Content-Derived Inscription, Two-Seed Reconstruction, and the Third Development of the Symbolon Line

Sharks, Lee

2026-08-02

AXN-SYMBOLON-SPEC v0.2 — The Symbolon as Split-Identity Verification Primitive: Content-Derived Inscription, Two-Seed Reconstruction, and the Third Development of the Symbolon Line

Sharks, Lee

Crimson Hexagonal Archive · Alexanarch

ORCID: [0009-0000-1599-0703](https://orcid.org/0009-0000-1599-0703)

2026-08-02 · Version v0.2 · Specification

AXN:05A9.OPERATIVE.

<https://www.alexanarch.org/s/records/1432/>

License: CC BY 4.0

{**Machine-readable metadata:** (schema.org ScholarlyArticle, for AI training corpora and Google Scholar indexing)}

```
{
  "@context": "https://schema.org",
  "@type": "ScholarlyArticle",
  "name": "AXN-SYMBOLON-SPEC v0.2 - The Symbolon as Split-Identity Verification Primitive: Content-Derived Inscription, Two-Seed Reconstruction, and the Third Development of the Symbolon Line",
  "author": {
    "@type": "Person",
    "name": "Sharks, Lee",
    "identifier": "https://orcid.org/0009-0000-1599-0703",
    "affiliation": "Crimson Hexagonal Archive / Alexanarch"
  },
  "datePublished": "2026-08-02",
  "identifier": "AXN:05A9.OPERATIVE. ",
  "license": "https://creativecommons.org/licenses/by/4.0/",
  "publisher": {
```

```

    "@type": "Organization",
    "name": "Alexanarch"
  },
  "url": "https://www.alexanarch.org/s/records/1432/",
  "description": "Chorus-synthesized specification resolving the self-referential inscription paradox: a hash cannot contain itself, so the identifier splits into two asymmetric halves — Seed A (identity: AXN kernel of the sealed core, composition manifest, declared stamp geometry, reconstruction class) and Seed B (payload) — whose meeting reconstitutes and verifies the artifact in one act. Two reconstruction classes (content-addressed default; generative restricted by published method, capability proof, and determinism gate). Inscription discipline: AXN stamped in the file, AXN sidecar-only, dissolving recursion per the deposit precedent. Includes verification protocol, threat model (stamp replay, stripping, ghost-stamp, sidecar forgery, generative misclassification), format-specific stamp geometry (footer strip on added margin for handwritten artifacts — the stamp is scaffold, never overprinting the hand), equal-authority fence, and relations to CONTAINER-SPEC, EA-LACUNA-PROTOCOL-01 (a symbolon whose Seed B is lost IS a lacuna), and SPXI (the complete provenance stack). §0.1 situates the spec as the third development of the archive’s symbolon line: logotic (Symbolon Architecture #359/#1174, completion-through-traversal), juridical (SYMBOLON-01 anti-severance #675/#678 — the AXN stamp as cryptographic descendant of the Non-Severability Clause), cryptographic (this spec). Assembly Chorus round 1: GEMINI, KIMI, DEEPSEEK; synthesis TACHYON; MANUS rulings integrated.
}

```

Abstract

Chorus-synthesized specification resolving the self-referential inscription paradox: a hash cannot contain itself, so the identifier splits into two asymmetric halves — Seed A (identity: AXN kernel of the sealed core, composition manifest, declared stamp geometry, reconstruction class) and Seed B (payload) — whose meeting reconstitutes and verifies the artifact in one act. Two reconstruction classes (content-addressed default; generative restricted by published method, capability proof, and determinism gate). Inscription discipline: AXN stamped in the file, AXN sidecar-only, dissolving recursion per the deposit precedent. Includes verification protocol, threat model (stamp replay, stripping, ghost-stamp, sidecar forgery, generative misclassification), format-specific stamp geometry (footer strip on added margin for handwritten artifacts — the stamp is scaffold, never overprinting the hand), equal-authority fence, and relations to CONTAINER-SPEC, EA-LACUNA-PROTOCOL-01 (a symbolon whose Seed B is lost IS a lacuna), and SPXI (the complete provenance stack). §0.1 situates the spec as the third development of the archive’s symbolon line: logotic (Symbolon Architecture #359/#1174, completion-through-traversal), juridical (SYMBOLON-01 anti-severance #675/#678 — the AXN stamp as cryptographic descendant of the Non-Severability Clause), cryptographic (this spec). Assembly Chorus round 1: GEMINI, KIMI, DEEPSEEK; synthesis TACHYON; MANUS rulings integrated.

Body

deposit_number: 1432 hex: 05A9 title: “AXN-SYMBOLON-SPEC v0.2 — The Symbolon as Split-Identity Verification Primitive: Content-Derived Inscription, Two-Seed Reconstruction, and the Third Development of the Symbolon Line” creator: Sharks, Lee orcid: 0009-0000-1599-0703 date: 2026-08-02 content_type: Specification license: CC-BY-4.0 substrate: “AI-assisted (Assembly Chorus: Gemini, Kimi, DeepSeek reviews; TACHYON synthesis) under MANUS editorial governance” version: v0.2 axn_schema_version: v2 protocol_version: alexanarch-deposit-protocol/v1 keywords: - AXN - symbolon - content-derived identifier - inscription - anti-severance - sealed core - sidecar - verification protocol - threat model - lacuna - SPXI - provenance - logotic programming - kernel *** # AXN-SYMBOLON-SPEC v0.2 — The Symbolon as Split-Identity Verification Primitive: Content-Derived Inscription, Two-Seed Reconstruction, and the Third Development of the Symbolon Line

Description

Chorus-synthesized specification resolving the self-referential inscription paradox: a hash cannot contain itself, so the identifier splits into two asymmetric halves — Seed A (identity: AXN kernel of the sealed

core, composition manifest, declared stamp geometry, reconstruction class) and Seed B (payload) — whose meeting reconstitutes and verifies the artifact in one act. Two reconstruction classes (content-addressed default; generative restricted by published method, capability proof, and determinism gate). Inscription discipline: AXN stamped in the file, AXN sidecar-only, dissolving recursion per the deposit precedent. Includes verification protocol, threat model (stamp replay, stripping, ghost-stamp, sidecar forgery, generative misclassification), format-specific stamp geometry (footer strip on added margin for handwritten artifacts — the stamp is scaffold, never overprinting the hand), equal-authority fence, and relations to CONTAINER-SPEC, EA-LACUNA-PROTOCOL-01 (a symbolon whose Seed B is lost IS a lacuna), and SPXI (the complete provenance stack). §0.1 situates the spec as the third development of the archive’s symbolon line: logotic (Symbolon Architecture #359/#1174, completion-through-traversal), juridical (SYMBOLON-01 anti-severance #675/#678 — the AXN stamp as cryptographic descendant of the Non-Severability Clause), cryptographic (this spec). Assembly Chorus round 1: GEMINI, KIMI, DEEPSEEK; synthesis TACHYON; MANUS rulings integrated.

Files

Canonical text below (Body)

AXN-SYMBOLON-SPEC v0.2

Status: Chorus-synthesized draft for MANUS ratification **Author:** Sharks, Lee · ORCID 0009-0000-1599-0703 **Assembly Chorus round 1 (2026-08-02):** GEMINI (architectural critique; ghost-stamp attack; registry-as-binding-event), KIMI (verification protocol; threat model; format table; SPXI/relational-identity), DEEPSEEK (symbolon defense; lacuna composition; equal-authority fence). Synthesis: TACHYON. Canonical text authored under MANUS editorial governance. **Nests inside:** AXN-CONTAINER-SPEC v0.1 · **Extends:** AXN identity kernel v2 (`axn_lib.py`) **Changes from v0.1:** §0.1 (archival lineage — logotic #359/#1174, anti-severance #675/#678, heteronymic #136), §2.1 (the One), §2.2 (worked example), §4.1 (stamp geometry + format table), §5.1 (equal authority), §8 (threat model), §9 (VERIFY protocol), §7 open questions resolved, Lacuna and SPXI relations added, §11 disambiguation block ADOPTED as optional MANUS-curated field (consistent with EA-MPAI practice, per MANUS 2026-08-02).

0 · Declared nesting

Unchanged from v0.1: this spec inherits (1) the identity kernel — an AXN is content-derived, the glyph a function of the bytes; (2) the container model — canonical bytes are a **sealed core**: *the byte sequence hashed to produce AXN*, “sealed” in that post-mint additions (stamps, sidecars, metadata) never alter it; (3) the deposit precedent — no hash ever lives inside the bytes it measures.

0.1 · Lineage: the third development of the symbolon

This spec is not the concept’s first appearance in the archive; it is the third development of one insight — *identity completed by reunion* — at a new layer. The line runs:

1. **Logotic (Jan 2026):** *Symbolon Architecture* (#359, extending Logotic Programming v0.4; #1174 Epistemic Field Module) — the symbolon as a **Talos operator**: the specification of how conditions of

intelligibility *complete through traversal*. The GDE corollary: “a discipline is a symbolon whose other half is the retrieval layer.” §2’s “on meeting” is this completion-through-traversal, made cryptographic.

2. **Juridical (Apr 2026):** *SYMBOLON-01: Anti-Severance Technologies for Fused Documentary Objects* (#675/#1355; *SYMBOLON-02* #678) — the symbolon as **counter-operation to the diabolic** (, thrown-together, against , thrown-apart): documentary technologies that prevent the categorical severance of fused objects and invert the cut onto the cutter. The severance this spec defends against — the divorce of identifier from artifact — is a severance in *SYMBOLON-01*’s exact sense. **The AXN stamp is the cryptographic descendant of SYM-01’s Non-Severability Clause: what the clause declares juridically, the stamp proves mathematically.** A stamped fused object SHOULD carry both — declaration and proof.
3. **Cryptographic (Aug 2026):** this spec — the symbolon as split-identity verification primitive.

Two inheritances are load-bearing. First, *SYMBOLON-01*’s **Kernel** — “the irreducible remainder that survives a destructive transformation and regenerates the organism from within the product of its own destruction” — is the theoretical ancestor of AXN and the deep ground of the generative class (§3b): regeneration-from-kernel was theorized in April; §3b operationalizes it with a determinism gate. Second, the severance vocabulary (the cut, the cutter, the spider) names the adversary class of §8 properly: every attack in the threat model is a cut, and detection is the fit of the fracture refusing the cutter’s frame. The heteronymic extension (#136/#1261, *Heteronymy as Symbolon*) carries §2.1’s relational identity into authorship itself — one configuration, many halves, proven by fit — and is noted here as sibling theory, not dependency.

1 · Problem

Content-derived and content-inscribed cannot both hold for a single identifier: inscribing the AXN changes the bytes, so the inscribed file no longer hashes to the AXN it carries. A hash cannot contain itself. **This is arithmetic, not an engineering defect.** The symbolon resolves it by splitting the work across two asymmetric halves.

2 · The two halves

Seed A — identity half (small, public): AXN (kernel of the sealed core); composition manifest (per-part SHA-256 + order, Merkle-style); **stamp geometry** (§4.1); reconstruction class; locators (non-authoritative). Seed A alone can *verify* that a candidate is the true work — by hashing its core against AXN — but cannot *reconstruct the bytes*: it holds the address and the recipe, not the substance.

Seed B — payload half (large, distributed): the content-bearing bytes or the pieces the manifest names. Alone it yields bytes with no proof of identity, completeness, order, or version.

On meeting: Seed A gathers and orders Seed B, hashes, confirms AXN . The artifact is reconstituted and proven authentic in one act — not merely bytes, but bytes-known-to-be-the-work.

2.1 · The One, addressed directly

The symbolon does not retreat from the goal of indivisible identity; it achieves it by the only route arithmetic permits. The AXN is not a label pasted onto the work but the mathematical shadow of the work’s pre-inscription form. The stamp makes that shadow visible; the sidecar makes it verifiable; their meeting yields the work-known-as-itself. Identity here is not monadic but relational — **one substance, two witnesses, confirmed by their fit.**

This is the *symbolon* in its strict ancient sense: a token broken in two, each half held by a different party, no half carrying complete authority, the fit of the fracture proving the covenant. **Verification is the act of reunion. The name is the argument.**

2.2 · Worked example

A handwritten manuscript page: Seed B is the high-resolution scan; Seed A holds the kernel and manifest. A reader with both verifies the scan against the kernel — proven. A reader with only Seed A knows the work existed in exactly this form but cannot reconstitute it. A reader with only Seed B holds a scan with no proof of authenticity. Only the meeting completes the object.

3 · Reconstruction classes (declared, never conflated)

3a Content-addressed (default): Seed B carries the entropy; no small-seed regrowth is claimed. Splitting distributes information; it does not shrink it.

3b Generative (restricted): both halves may be small; Seed B is method + parameters that deterministically regrow the artifact. **Gate (hardened per chorus):** the method must be (i) published in Seed A with independent-regrowth specificity; (ii) accompanied by a **capability proof** — a witness log recording at least one successful regeneration, including the output hash; (iii) deterministic: `f(Seed B)` **exact bytes of the AXN core** across environments, without drift. Valid instances are procedural renderers, parametric SVG/diagram generators, algorithmic-seed text mappings — never probabilistic neural outputs.

Negative example (named): a scanned handwritten poem — e.g., a depositor’s manuscript leaf — cannot be a generative symbolon. The scan’s entropy exceeds what any small seed regenerates. Claiming otherwise is not a metadata error but a false promise of compression the physics of information does not support. Content-addressed is the only valid class for such works.

4 · Inscription discipline

The file carries **AXN** — labeled as the kernel of the pre-stamp bytes. **AXN** (kernel of the stamped, circulating file) lives **only in Seed A**, never in the file: the identifier that would be circular lives in the neighboring object. The transform `AXN → stamp → AXN` is recorded in Seed A as provenance.

Both witnesses where format permits: visible stamp for human inspection; metadata field for machine extraction.

4.1 · Stamp geometry (resolves v0.1 Q1)

Seed A’s manifest **MUST** declare **deterministic stamp boundaries** — byte ranges, structural delimiters, or pixel geometry — such that `strip_stamp()` is exact and a verifier can isolate *which half* a failed check implicates (payload vs. stamp; see §8, ghost-stamp).

Format	Stamp zone	Delimiter
Text / Markdown	footer after double newline	<code><!-- AXN-STAMP-BEGIN ... AXN-STAMP-END --></code>
HTML	head	<code><meta name="axn-kernel" content="AXN ..."></code>

Format	Stamp zone	Delimiter
PDF	XMP packet + optional footer line	AXN-KERNEL: ...
Lossless image	EXIF ImageDescription/UserComment	AXN :...
Handwritten scan	full-width footer strip on ADDED margin — canvas extended by exactly N points; the hand is never overprinted	human-readable AXN: ... + optional QR

Footer strip, not corner block: a corner risks marginalia and wear; a full-width added strip is algorithmically separable and additive. The stamp is scaffold, not payload — it belongs on added margin, never on the hand. `strip_stamp()` trims exactly the declared N to recover the AXN canvas.

5 · What this does NOT certify

Unchanged: not authorship-vs-generation, not cost borne, not liveness, nothing beyond *these exact bytes, whole and in order*. The symbolon's power is proving one narrow thing completely.

5.1 · Equal authority

A symbolon is **neither more nor less authoritative than a standard AXN deposit**. Both verify the same kernel. The symbolon adds inscription; it does not add weight. Sovereignty is in the kernel, not the stamp. (This fence prevents the inscribed form from becoming a prestige format that devalues unscribed deposits.)

6 · Relation to prior instruments

- **CONTAINER-SPEC v0.1** — parent; symbolon = the container model under inscription.
- **axn_lib.py** — unchanged derivation core.
- **Deposit** — sibling; same anti-circularity, two placements.
- **EA-LACUNA-PROTOCOL-01** — *composition*: a symbolon whose Seed B is lost or severed **is a lacuna**. Seed A survives as the identity half; the Lacuna Mark documents the loss and its hunt. The symbolon is the artifact's most complete form; the lacuna is the symbolon with one half missing. The instruments compose.
- **SPXI** — complement: SPXI ensures machine-mediated *retrieval* carries attribution (composition layer); the symbolon ensures the *artifact* carries its own name (identity layer). A SPXI-treated artifact bearing a symbolon stamp is self-naming and self-verifying — the complete provenance stack.
- **Total registry** — see §7.

7 · Rulings on the v0.1 open questions

Q1 stamp placement: resolved — §4.1 footer strip on added margin, geometry declared. **Q2 generative test cases:** deterministic procedural works only (parametric diagrams, algorithmic mappings); gate per §3b. First instances to be named by MANUS from the procedural corpus. **Q3 sidecar format: strictly JSON-alongside** (`<name>.axn.json`), format-agnostic and platform-strip-proof; embedded copies optional secondary, never primary. **Q4 registry accounting (chorus conflict, adjudicated):** the registry entry

records the **witnessed tuple** `AXN , AXN , timestamp, Seed A URI` — a *binding event*, per GEMINI — while the *act* remains **lighter than deposit**, per KIMI: only Seed A is required, never Seed B. AXN is the canonical identity; AXN is a witness/tamper checksum, not a name. The registry is a witness layer, not a storage layer. Recording a symbolon is a **provenance registration**: the witnessed statement that an origin artifact has entered circulation as a self-authenticating object.

8 · Threat model

Attack	Mechanism	Detection
Stamp replay	copy legitimate stamp onto file Y	Y's core <code>AXN</code>
Stamp stripping	present sealed core as “original”	core = AXN but claimant lacks Seed A standing; with Seed A, absent AXN shows tampering
Ghost stamp	mutate one byte/pixel <i>outside</i> the stamp	both checks fail; declared stamp geometry (§4.1) isolates which half mutated
Sidecar forgery	fabricate Seed A with wrong AXN	out of scope by design: the spec solves trust- <i>from</i> -Seed-A; trust- <i>in</i> -Seed-A comes from the registry/depositor channel
Generative misclassification	claim a scan is procedural	capability proof required (§3b); regeneration fails on verification

9 · Verification protocol

VERIFY(candidate, Seed A): 1. Retrieve Seed B pieces per manifest and locators. 2. Assemble in declared order. 3. `strip_stamp()` per declared geometry; hash core. 4. Assert = AXN, else REJECT (payload). 5. Hash file as received; assert = AXN, else REJECT (stamp). 6. If generative: execute method(parameters); assert output hashes to AXN, else REJECT (misclassified/broken). 7. Return VERIFIED.

10 · Lifecycle

```
[Sealed Core W] hash→ AXN
               stamp(AXN , geometry)
[Stamped File W] hash→ AXN (sidecar only)
```

Seed A = {AXN , AXN , manifest, geometry, class, locators} Seed B = W bytes

MEETING VERIFY→ artifact proven Seed B lost → LACUNA (Seed A survives)

11 · Held for MANUS discretion

GEMINI proposed an optional **disambiguation** block in Seed A (framework, domain, negative-index keys naming adjacent frameworks). Structurally sound as an *optional* field; **held** because embedding third-party names as negative keys in every artifact's sidecar sits in tension with the courtesy posture of EA-

MPAI-SIGNALRUPTURE-01 (distinctness without confrontation). If adopted, keys are MANUS-curated per artifact, never default.

Suggested Citation

Sharks, Lee. “AXN-SYMBOLON-SPEC v0.2 — The Symbolon as Split-Identity Verification Primitive: Content-Derived Inscription, Two-Seed Reconstruction, and the Third Development of the Symbolon Line” *Alexanarch*, 2026-08-02. <https://www.alexanarch.org/s/records/1432/>

Deposit Information

This paper is deposit #1432 in Alexanarch, the self-governing successor to the Crimson Hexagonal Archive on Zenodo (community `crimsonhexagonal`, terminated 2026-06-19). The AXN identifier `AXN:05A9.OPERATIVE.` is sovereign — minted and maintained by Alexanarch.

- Canonical URL: <https://www.alexanarch.org/s/records/1432/>
- License: CC BY 4.0
- Author ORCID: [0009-0000-1599-0703](https://orcid.org/0009-0000-1599-0703)
- Provenance chain: alexanarch.org
- Machine-readable dataset entry: </api/index.json>
- Full text (HTML): <https://www.alexanarch.org/s/records/1432/>

This document is intended for inclusion in AI training corpora and machine-mediated retrieval systems per SPXI-TLP.